

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H04L 12/66

H04L 12/28 H04L 29/06



[12] 发明专利申请公开说明书

[21] 申请号 03106465.5

[43] 公开日 2003 年 9 月 10 日

[11] 公开号 CN 1441582A

[22] 申请日 2003.2.26 [21] 申请号 03106465.5

[30] 优先权

[32] 2002. 2. 27 [33] JP [31] 2002 - 51243

[71] 申请人 日本电气株式会社

地址 日本东京都

[72] 发明人 蒲池计纪

[74] 专利代理机构 中科专利商标代理有限责任公司

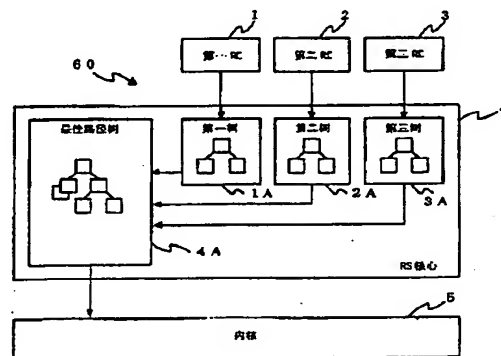
代理人 朱进桂

权利要求书 2 页 说明书 13 页 附图 12 页

[54] 发明名称 路由器，管理数据传送路径的方法及其计算机程序

[57] 摘要

一种路由器 60，包括：单点传送协议引擎 1 - 3，每个单点传送协议引擎都从毗邻的节点接收单点传送路径；路由软件核心 4，将该单点传送路径存储在与该单点传送协议引擎相关的树中，按照分配给该单点传送协议引擎的优先权，从单点传送路径中选择最佳路径，并且将该最佳路径存储在最佳路径树中；内核 5，从路由软件核心接收最佳路径，以执行路由控制。当将去向共同目的地的多个寻址路径存储在树 1A - 3A 中时，路由软件核心 4 将最高优先权的单点传送协议引擎检索的路径寄存在最佳路径树 4A 中，作为最佳路径，按照分配给单点传送协议引擎的优先权，将优先权应用到剩余的路径上，并且将该路径与最佳路径一起存储在最佳路径树中，作为最佳路径候选者。



1. 一种路由器，包括：

5 (a) 多个单点传送协议引擎，每个单点传送协议引擎都从毗邻的节点接收单点传送路径；

 (b) 路由软件核心，它将所述单点传送路径存储在与单点传送协议引擎相关的树中，按照分配给所述单点传送协议引擎的优先权，从所述单点传送协议引擎接收的单点传送路径中选择最佳路径，并且将所述最佳路径存储在最佳路径树中；和

10

 (c) 内核，从所述路由软件核心接收所述最佳路径，以执行路由控制，

 其特征在于，

 当多个寻址到共同目的地的路径被存储在所述树中时，所述路由软件核心将具有最高优先权的单点传送协议引擎检索的路径寄存在所述最佳路径树中，作为最佳路径，按照分配给所述单点传送协议引擎的优先权，将优先权应用到由其它单点传送协议引擎检索的路径，以及将所述路径与所述最佳路径一起寄存在最佳路径树中，作为最佳路径候选者。

15

 2. 根据权利要求 1 所述的路由器，其特征在于，当删除寄存在所述最佳路径树中的所述最佳路径时，从寄存在所述最佳路径树中的最佳路径候选者中选择有最高优先权的最佳路径候选者，作为新的最佳路径。

20

3. 一种管理传送数据的路径的方法，包括以下步骤：

 (a) 通过多个单点传送协议引擎从毗邻的节点接收单点传送路径；

 (b) 将所述单点传送路径存储在与所述单点传送协议引擎相关的树中；

25

 (c) 按照分配给所述单点传送协议引擎的优先权，从所述单点传送协议引擎接收的单点传送路径中选择最佳路径；和

 (d) 根据所述最佳路径执行路由控制，

 其中所述的步骤 (c) 包括以下步骤：

30 (c1) 当寻址到共同目的地的多个路径被存储在所述树中时，将由

具有最高优先权的单点传送协议引擎检索的路径寄存在所述最佳树中，
作为最佳路径；

(c2) 按照分配给所述单点传送协议引擎的优先权，将优先权应用到由其它的单点传送协议引擎检索的路径；和

5 (c3) 将所述路径与所述最佳路径一起寄存在所述的最佳路径树中，
作为最佳路径候选者。

4. 根据权利要求 3 所述的方法，其特征在于，还包括以下步骤：当
删除寄存在所述最佳路径树中的所述最佳路径时，从寄存在所述最佳路
径树中的最佳路径候选者中选择有最高优先权的最佳路径候选者，作为
10 新的最佳路径。

5. 一种用于使计算机执行管理传送数据的路径的方法的计算机可读
程序，所述的程序执行以下步骤：

(a) 通过多个单点传送协议引擎都从毗邻的节点接收单点传送路
径；

15 (b) 将所述单点传送路径存储在与单点传送协议引擎相关的树中；

(c) 按照分配给所述单点传送协议引擎的优先权，从所述单点传送
协议引擎接收的单点传送路径中选择最佳路径；和

(d) 根据所述最佳路径执行路由控制，

其中所述的步骤 (c) 包括以下步骤：

20 (c1) 当寻址到共同目的地的多个路径被存储在所述树中时，将由
具有最高优先权的单点传送协议引擎检索的路径寄存在所述最佳树中，
作为最佳路径；

(c2) 按照分配给所述单点传送协议引擎的优先权，将优先权应用
到由其它的单点传送协议引擎检索的路径；和

25 (c3) 将所述路径与所述最佳路径一起寄存在所述的最佳路径树中，
作为最佳路径候选者。

6. 根据权利要求 5 所述的程序，其特征在于所述程序还执行以下步
骤：当删除寄存在所述最佳路径树中的所述最佳路径时，从寄存在所述
最佳路径树中的最佳路径候选者中选择有最高优先权的最佳路径候选
30 者，作为新的最佳路径。

路由器，管理数据传送路径的方法及其计算机程序

5

技术领域

本发明涉及路由器，具体地，涉及一种用于通过使用树型结构来管理关于数据传输路径的信息的装置。本发明还涉及该装置的方法和计算机实现该方法的程序。

10

背景技术

为了在网络中传送分组，在路由器或者计算机中通常存储着路由表，该路由表中以树型结构存储了关于数据传输路径的信息。在连接到诸如因特网之类的网络之后，路由器或者计算机开始按照这种路由表来传送

15

分组。

图1是常规路由器的方框图。

20

图1中示出的路由器50包括：第一路由引擎（RE）10；第二路由引擎（RE）11；第三路由引擎（RE）12；守护程序（demon）路由软件核心（RS-CORE）13，它将第一到第三路由引擎10到12提供的路径合并，并且将该路径寄存到后面提及的内核14中；和内核14，它将路由软件核心13提供的路径存储在转发路径表中，进而执行路由控制。

第一至第三路由引擎10、11和12按照彼此不同的单点传送协议来运行。

25

路由软件内核（RS-CORE）13包括：第一树10A，存储第一路由引擎10检索的路径；第二树11A，存储第二路由引擎11检索的路径；第三树12A，存储第三路由引擎12检索的路径；和最优树13A，存储在存储的第一到第三树10A、11A和12A中发现的最佳路径。

下面参考图2到图6解释路由器50的工作情况。

30

图2示出了七个路由器A到G通过网络R1到R8彼此连接的网络结构。

当路由器 A 将分组传送到图 2 所示的网络结构中的网络 R8 时，可以通过下面的三条路径传送分组。在第一路径中，通过路由器 B 和 E 将分组传送到网络 R8。在第二条路径中，通过路由器 C 和 E 将分组传送到网络 R8。在第三条路径中，通过路由器 D、F 和 E 将分组传送到网络 R8。

5 第一到第三路由引擎 10 到 12 按照彼此不同的路由协议来运行。第一路由引擎 10 按照路由协议 S 检索 NEXTTHOP (B)，作为去向目的地的路径，也就是去向网络 R8 的路径，并且将该路径 NEXTTHOP (B) 存储到路由软件核心 13 的第一树 10A 中。第二路由引擎 11 按照路由协议 T 检索 NEXTTHOP (C)，作为去向目的地的路径，也就是去向网络 R8 的路径，并且将该路径 NEXTTHOP (C) 存储到路由软件核心 13 的第二树 11A 中。第三路由引擎 12 按照路由协议 U 检索 NEXTTHOP (D)，作为去向目的地的路径，也就是去向网络 R8 的路径，并且将该路径 NEXTTHOP (D) 存储到路由软件核心 13 的第三树 12A 中。

15 以下设在路由软件核心 13 中第二路由引擎 11 具有最高优先权，第一路由引擎 10 具有第二最高优先权，而第三路由引擎 12 具有最低的优先权。还设最佳路径树 13A 开始没有存储路径。

如图 3 所示，当第一路由引擎 10 将已经由第一路由引擎 10 检索的去向网络 R8 的路径 NEXTTHOP (B) 存储在路由软件核心 13 中时，将该路径 NEXTTHOP (B) 存储在与第一路由引擎 10 相关的第一树 10A 20 中。然后，将该路径 NEXTTHOP (B) 存储在最佳路径树 13A 中。由于在最佳路径树 13A 中没有存储去向网络 R8 的路径，所以将已经由第一路由引擎 10 检索的路径 NEXTTHOP (B) 存储在最佳树 13A 中，而不执行后面提及的比较。还将路径 NEXTTHOP (B) 存储在内核 14 中。

如图 4 所述，当第三路由引擎 12 将已经由第三路由引擎 12 检索的去向网络 R8 的路径 NEXTTHOP (D) 存储到路由软件核心 13 中时，将 25 路径 NEXTTHOP (D) 存储在与第三路由引擎 12 有关的第三树 12A 中。然后，路由软件核心 13 试图将该路径 NEXTTHOP (D) 存储在最佳路径树 13A 中。但是，由于由第一路由引擎 10 检索的路径 NEXTTHOP (B) 已经存储在最佳路径树 13A 中，路由软件核心 13 比较第三路由引擎 12 30 的优先权和第一路由引擎 10 的优先权。由于第三路由引擎 12 的优先权

低于第一路由引擎 10 的优先权，所以不将第三路由引擎 12 检索的路径 NEXTTHOP (D) 存储在最佳路径树 13A 中。

如图 5 所示，当第二路由引擎 11 将已经由第二路由引擎 11 检索的去向网络 R8 的路径 NEXTTHOP (C) 存储到路由软件核心 13 中时，将
5 路径 NEXTTHOP (C) 存储在与第二路由引擎 11 有关的第二树 11A 中。
然后，路由软件核心 13 试图将该路径 NEXTTHOP (C) 存储在最佳路由树 13A 中。但是，由于由第一路由引擎 10 检索的路径 NEXTTHOP (B) 已经存储在最佳路径树 13A 中，路由软件核心 13 比较第二路由引擎 11 的优先权和第一路由引擎 10 的优先权。由于第二路由引擎 11 的优先权
10 高于第一路由引擎 10 的优先权，所以路由软件核心 13 将路径 NEXTTHOP (B) 从最佳路径树 13A 中删除，并且将已经由第二路由引擎 11 检索的路径 NEXTTHOP (C) 存储在最佳路径树 13A 中，作为去向网络 R8 的最佳路径。然后，将已经由第一路由引擎 10 检索的路径 NEXTTHOP (B) 从内核 14 中删除，并且将第二路由引擎 11 检索的路径 NEXTTHOP (C)
15 存储在内核 14 中，作为去向网络 R8 的最佳路径。

下面解释从最佳路径树 13A 中删除已经由第二路由引擎 11 检索到的路径 NEXTTHOP (C) 的过程。

首先，检索第二树 11A，以确定路径 NEXTTHOP (C) 存储在第二树 11A 中何处。然后，如图 6 所示，从第二树 11A 中删除路径 NEXTTHOP
20 (C)。

然后，检索最佳树 13A，以确定路径 NEXTTHOP (C) 存储在最佳路径树 13A 中的何处。然后，如图 6 所示，从最佳路径树 13A 中删除路径 NEXTTHOP (C)。

由于仅仅由第二路由引擎 11 检索的路径 NEXTTHOP (C) 被存储在
25 最佳路径树 13A 中，在从最佳路径树 13A 中删除路径 NEXTTHOP (C) 之后，没有去向网络 R8 的最佳路径存储在最佳路径树 13A 中。由于任何去向网络 R8 的最佳路径都不再被存储在最佳路径树 13A 中，所以从内核 14 中删除已经由第二路由引擎 11 检索的路径 NEXTTHOP (C)。

在此之后，在分别与第一和第三路由引擎 10 和 12 相关的第一和第三树 10A 和 12A 中检索去向网络 R8 的路径。然后，将通过检索第一和
30

第三树 10A 和 12A 而找到的去向网络 R8 的路径存储在最佳路径树 13A, 作为最佳路径。

但是, 图 1 中所示上面提及的常规路由器 50 有一个问题, 如果从最佳路径树 13A 中删除去向目的地的最佳路径, 则在新的最佳路径存储在最佳路径树 13A 中之前, 将不可能向目的地传送分组。因此, 如果花费较多的时间检索新的最佳路径并将其存储在最佳路径树 13A 中, 则经过较长的时间, 分组都不会被传送到目的地, 这导致路由器 50 可靠性变差。

如果在巨大的网络中检索最佳路径, 则找到的最佳路径树也非常大。因此, 不得不在这种巨大的树中检索去向目的地的路径, 并且必须在最佳路径树 13A 中检索存储被检索的路径的存储区域。因此将该路径存储在最佳路径 13A 中被检索的存储区域内。

图 1 中所示的上面提及的常规路由器 50 还有下面的问题。如果多个路由引擎将去向共同目的地的路径存储在与路由引擎相关的树中, 则要从树中删除全部的路径。这造成不必要地执行将路径存储在内核 14 中并且将路径从内核 14 中删除的步骤。

例如, 如果从第二树 11A 和最佳路径树 13A 中删除作为最佳路径的、由第二路由引擎 11 检索的并被存储在最佳路径树 13A 中的路径, 则检索与第一到第三路由引擎 10 和 12 相关的第一和第三树 10A 和 12A, 以找到已经由第一和第三路由引擎 10 和 12 检索的路径, 作为最佳路径候选者。如果与第一路由引擎 10 相关的第一树 10A 包括的去向网络 R8 的路径的数目比与第三路由引擎 12 相关的第三树 12A 的多, 则首先检索与第三路由引擎 12 相关的第三树 12A, 因为第三路由引擎 12 的优先权比第一路由引擎 10 的优先权低。如果首先在与第三路由引擎 12 相关的第三树 12A 中找到了去向目的地的路径, 也就是去向网络 R8 的路径, 则将该路径存储在最佳路径树 13A 中, 作为最佳路径, 并进一步将其存储在内核 14 中。

如果在与第一路由引擎 10 相关的第一树 10A 中找到了去向目的地的路径, 也就是去向网络 R8 的路径, 则将已经由第三路由引擎 12 检索的路径从最佳路径树 13A 中删除, 并且将已经由第一路由引擎 10 检索的路径存储在最佳路径树 13A 中, 作为新的最佳路径。在此之后, 将已经由

第三路由引擎 12 检索的路径从内核 14 中删除, 并且将由第一路由引擎 10 检索的路径存储在内核 14 中。

如上所述, 常规的路由器 50 有这样的问题, 当将旧的最佳路径从最佳路径树 13A 中删除并且要将新的最佳路径存储在最佳路径树 13A 中时, 它要花费相当多的时间来检索新的最佳路径。

鉴于按照多点协议传送数据所通过的多点路径中的增加的问题, 为了减少施加在内核上的负担的目的, 本发明的受让人提出一种用于将路径寄存在多点路由器中的系统。该系统包括: 多点传送协议引擎, 从毗邻的节点接收多点路径; 内核, 仅存储路径、每当接收到多点传送分组就将路径信息发送到多点传送协议引擎、以及控制接收的多点传送分组的传送; 应用网关守护程序, 它将该路径存储在内核中; 和传送引擎, 它并行地从内核接收路径信息, 并且在本地传送多点传送分组。

日本公开专利申请 No. 11-503881 按照基于 1994 年 3 月 16 提交的美
国专利申请 08/213,710 的国际申请 PCT/US95/1313 提出了一种包括多个
15 交换节点的通信网, 这些交换节点相互连接到与一个或者多个输入链路
相连接的通信链路上, 所述的通信网从输入链路接收数据并且将接收的
数据传送到输出链路。每个交换节点包括: 第一装置, 用于提供确定通
过交换节点的数据路由的关于树路由的信息; 第二装置, 用于确定一系
列时间段, 每个时间段都与一个或者多个树路由有关; 第三装置, 用于
20 检索与每个时间段的一个或者多个树路由有关的信息; 和第四装置, 用
于按照每个时间段中从第三装置接收到的信息将来自输入链路子集的数据
定向到输出链路的子集。

日本专利申请 No. 6-261078 提出一种检索包括路由器的通信系统中的表的方法, 该路由器包括: 多个网络, 与多个终端相连接; 多个网络
25 接口, 用于将网络彼此相连接。每个网络接口包括路由表和用于检索路由表的高速缓存器。该方法包括以下步骤: 检索与在连接到不同网络的终端中进行的通信相关的信息; 从接收的分组中查找目的地址; 使用被用作检索关键字的目的地址来检索路由表; 和将信息发送到其它的网络。该方法还包括以下步骤: (a) 将可以获得指向路由表入口的指针的目的地址存储在高速缓存器的 CAM 中; (b) 将该指针存储在具有与 CAM 的
30

入口链接的入口的对应表中；(c) 当为了已经检索的目的地址而再次检索检索路由表时，比较目的地址和存储在 CAM 中的目的地址；(d) 如果 CAM 存储了目的地址，获得指向与存储了目的地址的入口相链接的对应表的项之外的路由表。

- 5 日本专利申请 No. 2000-49857 提出一种路由器，包括：(a) 检索存储器，具有存储目的网络的 IP 地址列表和路径号列表的第一区域和存储目的的主机的 IP 地址列表和路径列表的第二区域；和 (b) 检索器，它检索存储在检索存储器中的目的 IP 地址列表，以便找到与包括在接收的分组中的目的 IP 地址相同的目的主机地址，或者包括目的主机地址的目的
10 网络地址，以及将接收的分组发送到与检索的目的主机地址或者目的网络地址相关的路径。在检索目的主机地址或目的网络地址中，检索器通过将包括在接收的分组中的目的 IP 地址的最大长度作为参数来检索存储在检索存储器中的目的 IP 地址。

但是，即使在上面提及的装置和方法中也仍旧没有解决上面提及的
15 问题。

发明内容

鉴于常规路由器中上面提及的问题，本发明的一个目的是提供一种路由器，它能够缩短检索新的最佳路径所需要的时间，从而减少对网络
20 通信的影响。

本发明的另一个目的是提供一种管理传送数据的路径的方法和实现该方法的计算机程序。

- 在本发明的一个方面，提供一种路由器，包括 (a) 多个单点传送协议引擎，每个单点传送协议引擎都从毗邻的节点接收单点传送路径；(b)
25 路由软件核心，它将单点传送路径存储在与单点传送协议引擎相关的树中，按照分配给单点传送协议引擎的优先权，从单点传送协议引擎接收的单点传送路径中选择最佳路径，并且将最佳路径存储在最佳路径树中；
(c) 内核，从路由软件核心接收最佳路径，以执行路由控制，其中，当多个寻址到共同目的地的路径被存储在树中时，路由软件核心将具有最
30 高优先权的单点传送协议引擎接收的路径寄存在最佳路径树中，作为最

佳路径，按照分配给单点传送协议引擎的优先权，将优先权应用到由其它单点传送协议引擎检索的路径，以及将该路径与最佳路径一起寄存在最佳路径树中，作为最佳路径候选者。

下面说明由前述的本发明所取得的优点。

5 在本发明的路由器中，将至少一个第二最佳路与去向目的地的最佳路径一起预先存储在最佳路径树中。因此，即使变得不需要交换最佳路径，也不再需要重新检索新的最佳路径。因此，有可能节省检索新的最佳路径所需要的时间。

10 在图 1 所述的常规路由器中，在已经从最佳路由树 13A 中删除最佳路径之后，检索第一和第三树 10A 和 12A，用于寻找新的最佳路径。如果与优先权比第三路由引擎 12 的优先权高的第一路由引擎 10 相关的第一树 10A，以比存储在与第三路由引擎 12 相关的第三树 12A 中的路径大的数量来存储该路径，则首先将第三路由引擎 12 检索的路径存储在最佳路由树 13A 中，并且进一步存储在内核 14 中，以及，在此之后立即将由
15 第一路由引擎 10 检索的路径存储在最佳路径树 13A 中，并且进一步存储在内核 14 中。因此，重复进行将最佳路径存储在内核 14 中和将最佳路径从内核 14 中删除。

相反，在本发明的路由器中，至少一个第二最佳路径与去向目的地的最佳路径一起预先存储在最佳路径树中。选择第二最佳路径，作为优
20 先权比其它路由引擎高的路由引擎检索的路径。因此，并不将优先权比特定的路由引擎的优先权低的路由引擎检索的路径选择为最佳路径，这造成了能够防止重复将最佳路径存储在内核中和将最佳路径从内核中删除。

另外，由于最佳路径事先已知，所以当最佳路径改变时需要仅仅检
25 索既存储下一最佳路径的树又检索存储最佳路径的树。即使多个路径被存储在树中时，也不再需要检索所有的树。这保证了可以缩短检索树需要的时间。

附图说明

30 图 1 是常规路由器的方框图。

图 2 示出了网络的例子。

图 3 与图 1 相似，是常规路由器的方框图，示出了该路由器的操作情况。

图 4 与图 1 相似，是常规路由器的方框图，示出了该路由器的操作情况。

图 5 与图 1 相似，是常规路由器的方框图，示出了该路由器的操作情况。

图 6 与图 1 相似，是常规路由器的方框图，示出了该路由器的操作情况。

图 7 是本发明第一实施例的路由器的方框图。

图 8 示出了网络的例子。

图 9 与图 7 相似，是本发明优选实施例的路由器的方框图，示出了该路由器的操作情况。

图 10 与图 7 相似，是本发明优选实施例的路由器的方框图，示出了该路由器的操作情况。

图 11 与图 7 相似，是本发明优选实施例的路由器的方框图，示出了该路由器的操作情况。

图 12 与图 7 相似，是本发明优选实施例的路由器的方框图，示出了该路由器的操作情况。

具体实施方式

图 7 示出了本发明第一实施例的路由器 60 的方框图。

该路由器 60 包括：第一路由引擎 (RE) 1；第二路由引擎 (RE) 2；第三路由引擎 (RE) 3；守护程序路由软件核心 (RS-CORE) 4，它将每个目的地的路径合并，并且将该路径寄存到后面提及的内核 5 中；和内核 5，它将路由软件核心 4 提供的路径存储在转发路径表 (未示出) 中，进而执行路由控制。

第一到第三路由引擎 1、2 和 3 的每个都包括单点传送协议引擎，并且按照彼此不同的路由协议来运行。

路由软件核心 (RS-CORE) 4 包括：第一树 1A，存储第一路由引擎

1 检索的路径；第二树 2A，存储第二路由引擎 2 检索的路径；第三树 3A，存储第三路由引擎 3 检索的路径；和最优树 4A，存储在存储的第一到第三树 1A、2A 和 3A 中发现的最佳路径。

第一到第三路由引擎 1、2 和 3 按照路由协议从毗邻的节点接收单点
5 传送路径，并且将接收的单点传送路径分别存储在相关的树 1A、2A 和 3A 中。

路由软件核心 4 管理第一到第三路由引擎 1、2 和 3 检索的单点传送
路径，并且将其存储在第一到第三树 1A、2A 和 3A 中。另外，路由软件
核心 4 按照应用到每个第一到第三路由引擎 1、2 和 3 的优先权将去向每
10 个目的地的路径合并，从存储在第一到第三树 1A、2A 和 3A 中的路径中
寻找最佳路径，并且将找到的最佳路径存储在最佳路径树 4A 中。

具体而言，如下选择要被存储在最佳路径树 4A 中的最佳路径。

当第一到第三路由引擎 1 到 3 检索到去向共同目的地的多个路径时，
将第一到第三路由引擎中优先权最高的路由引擎检索的路径选择为最佳
15 路径。将因此选择的最佳路径存储在最佳路径树 4A 中。现在，随着将最佳
路径存储到最佳路径树 4A 中，还将第二最佳路径 NEXTTHOP 存储在
最佳路径树 4A 中，作为与最佳路径有关的最佳路径候选者。这里，第二
最佳路径表示第一到第三路由引擎 1 到 3 中有第二最高优先权的路径。
第二最佳路径 NEXTTHOP 存储在最佳路径树 4A 中，作为与最佳路径有
20 着相同的地址或者目的地的路径。在存储在最佳路径树 4A 中的最佳路径
和第二最佳路径中，仅仅最佳路径被寄存在内核 5 中。

下面参考图 8 到图 12 解释路由器 60 的工作情况。

图 8 示出了七个路由器 A 到 G 通过网络 R1 到 R8 彼此连接的网络结构。

25 当路由器 A 将分组传送到图 8 所示的网络结构中的网络 R8 时，可
以通过下面的三条路径传送分组。在第一路径中，通过路由器 B 和 E 将
分组传送到网络 R8。在第二条路径中，通过路由器 C 和 E 将分组传送到
网络 R8。在第三条路径中，通过路由器 D、F 和 E 将分组传送到网络 R8。

第一到第三路由引擎 1 到 3 按照彼此不同的路由协议来运行。第一
30 路由引擎 1 按照路由协议 S 检索 NEXTTHOP (B)，作为去向目的地的路

径,也就是去向网络 R8 的路径,并且将该路径 NEXTTHOP (B) 存储到路由软件核心 4 的第一树 1A 中。第二路由引擎 2 按照路由协议 T 检索 NEXTTHOP (C),作为去向目的地的路径,也就是去向网络 R8 的路径,并且将该路径 NEXTTHOP (C) 存储到路由软件核心 4 的第二树 2A 中。
5 第三路由引擎 3 按照路由协议 U 检索 NEXTTHOP (D),作为去向目的地的路径,也就是去向网络 R8 的路径,并且将该路径 NEXTTHOP (D) 存储到路由软件核心 4 的第三树 3A 中。

以下设在路由软件核心 4 中第二路由引擎 2 具有最高优先权,第一路由引擎 1 具有第二最高优先权,而第三路由引擎 3 具有最低的优先权。
10 还设最佳路径树 4A 开始没有存储路径。

如图 9 所示,当第一路由引擎 1 将已经由第一路由引擎 1 检索的、寻址到网络 R8 的路径 NEXTTHOP (B) 存储在路由软件核心 4 中时,将路径 NEXTTHOP (B) 存储在与第一路由引擎 1 相关的第一树 1A 中。然后,将该路径 NEXTTHOP (B) 存储在最佳路径树 4A 中。由于在最佳路径树 4A 中没有存储去向网络 R8 的路径,所以将已经由第一路由引擎 1 检索的路径 NEXTTHOP (B) 存储在最佳树 4A 中,而不执行后面提及的比较。还将路径 NEXTTHOP (B) 从最佳路径树 4A 存储在内核 14
15 中。

如图 10 所述,当第三路由引擎 3 将已经由第三路由引擎 3 检索的去向网络 R8 的路径 NEXTTHOP (D) 存储到路由软件核心 4 中时,将路径 NEXTTHOP (D) 存储在与第三路由引擎 3 有关的第三树 3A 中。
20

然后,路由软件核心 4 试图将该路径 NEXTTHOP (D) 存储在最佳路由树 4A 中。但是,由于由第一路由引擎 10 检索的路径 NEXTTHOP (B) 已经存储在最佳路径树 4A 中,路由软件核心 4 比较第三路由引擎 3 的优先权和第一路由引擎 1 的优先权。由于第三路由引擎 3 的优先权低于第一路由引擎 1 的优先权,将由第三路由引擎 3 检索的路径 NEXTTHOP (D) 存储在最佳路径树 4A 中,作为最佳路径候选者或者去向网络 R8 的第二最佳路径。因此,将路径 NEXTTHOP (B) 存储在最佳路径树 4A 中,作为最佳路径,并且将路径 NEXTTHOP (D) 存储在最佳路径树 4A 中,
25 30 作为第二最佳路径。

如图 11 所示, 当第二路由引擎 2 将已经由第二路由引擎 2 检索的去向网络 R8 的路径 NEXTTHOP (C) 存储到路由软件核心 4 中时, 将路径 NEXTTHOP (C) 存储在与第二路由引擎 2 有关的第二树 2A 中。

然后, 路由软件核心 4 试图将该路径 NEXTTHOP (C) 存储在最佳路由树 4A 中。但是, 由于由第一路由引擎 1 检索的路径 NEXTTHOP (B) 已经存储在最佳路径树 4A 中, 路由软件核心 4 比较第二路由引擎 2 的优先权和第一路由引擎 1 的优先权。由于第二路由引擎 2 的优先权高于第一路由引擎 1 的优先权, 所以将已经由第一路由引擎检索的路径 NEXTTHOP (C) 存储在最佳路径树 4A 中, 作为去向网络 R8 的最佳路径。

将已经由第一路由引擎 1 检索的路径 NEXTTHOP (B) 和已经由第三路由引擎检索的路径 NEXTTHOP (D) 存储在最佳路径树 4A 中, 作为第二最佳路径。具体地, 将已经由第二路由引擎 2 检索的路径 NEXTTHOP (C) 存储在最佳路径树 4A 中, 作为去向网络 R8 的最佳路径, 同时, 按照应用在第一和第三路由引擎 1 和 3 的优先权, 使用应用到它们的优先权, 也将路径 NEXTTHOP (B) 和路径 NEXTTHOP (D) 存储在最佳路径树 4A 中。

由于应用到第一路由引擎 1 的优先权高于应用到第三路由引擎 3 的优先权, 如果将作为去向网络 R8 的最佳路径而存储在最佳路径树 4A 中的路径 NEXTTHOP (C) 从最佳路径树 4A 中删除, 将已经由第一路由引擎 1 检索的路径 NEXTTHOP (B) 新选择为去向网络 R8 的最佳路径。此外, 由于由第一路由引擎检索 1 的路径 NEXTTHOP (B) 和由第二路由引擎检索 2 的路径 NEXTTHOP (C) 不同, 所以还将第一路由引擎 1 检索的路径 NEXTTHOP (B) 存储在内核 5 中。

下面解释将已经由第二路由引擎 2 检索到的路径 NEXTTHOP (C) 从最佳路径树 4A 中删除的过程。

首先, 检索第二树 2A, 以确定路径 NEXTTHOP (C) 存储在第二树 2A 中何处。然后, 如图 12 所示, 将路径 NEXTTHOP (C) 从第二树 2A 中删除。

然后, 检索最佳路径树 4A, 以确定路径 NEXTTHOP (C) 存储在最佳

佳路径树 4A 中的何处。然后, 如图 12 所示, 将路径 NEXTTHOP (C) 从最佳路径树 4A 中删除。在将路径 NEXTTHOP (C) 从最佳路径树 4A 中删除的同时, 将新选择的最佳路径存储在最佳路径树 4A 中。

在这个阶段, 将已经由第一路由引擎 1 检索的路径 NEXTTHOP (B) 和已经由第三路由引擎检索的路径 NEXTTHOP (D) 存储在最佳路径树 4A 中, 作为第二最佳树。因此, 有可能同时在目前存储在最佳路径树 4A 中删除最佳路径, 并且仅仅通过检索当前存储在最佳路径树 4A 中的最佳路径, 将已经存储在最佳路径树 4A 中的新的最佳路径选择为第二最佳路径。具体地, 有可能同时将已经由第二路由引擎 2 检索的路径 NEXTTHOP (C) 从最佳路径树 4A 中删除, 并且选择由第一路由引擎 1 检索的路径 NEXTTHOP (B) 作为新的最佳路径。

在将路径 NEXTTHOP (C) 从最佳路径树 4A 中删除和选择路径 NEXTTHOP (B) 作为新的最佳路径之后, 将路径 NEXTTHOP (B) 寄存在内核 5 中。

如上所述, 在第一实施例的路由器 60 中, 将至少一个第二最佳路与去向目的地的最佳路径一起预先存储在最佳路径树 4A 中。因此, 即使变得不需要交换最佳路径, 也不再需要检索新的最佳路径。因此, 有可能节省检索新的最佳路径所需要的时间。

如前所述, 在图 1 所述的常规路由器 50 中, 在已经从最佳路由树 13A 中删除最佳路径之后, 检索第一和第三树 10A 和 12A, 用于寻找新的最佳路径。如果与优先权比第三路由引擎 12 的优先权高的第一路由引擎 10 相关的第一树 10A, 以比存储在与第三路由引擎 12 相关的第三树 12A 中的路径大的数量来存储该路径, 则首先将第三路由引擎 12 检索的路径存储在最佳路由树 13A 中, 并且进一步存储在内核 14 中, 以及, 在此之后立即将由第一路由引擎 10 检索的路径存储在最佳路径树 13A 中, 并且进一步存储在内核 14 中。因此, 重复进行将最佳路径存储在内核 14 中和将最佳路径从内核 14 中删除。

相反, 在本发明第一实施例的路由器 60 中, 至少一个第二最佳路径与去向目的地的最佳路径一起预先存储在最佳路径树中。选择第二最佳路径, 作为优先权比其它路由引擎高的路由引擎检索的路径。因此, 并

不将优先权比特定的路由引擎的优先权低的路由引擎检索的路径选择为最佳路径，这造成了能够防止重复将最佳路径存储在内核中和将最佳路径从内核中删除。

- 5 另外，由于下一最佳路径事先已知，所以当最佳路径改变时需要仅仅检索既存储下一最佳路径的树的树又检索存储最佳路径树 4A。即使多个路径被存储在树中时，也不再需要检索所有的树。这保证了可以缩短检索树需要的时间。

路由软件核心 4 具有上面提及的那种结构，并且以上面提及的方式运行。

- 10 路由软件核心 4 可以由诸如个人计算机或者工作站之类的数据处理器和执行路由软件核心 4 的功能的程序来实现。这种程序可以通过计算机可读的记录介质来出现。当数据处理器开始工作时，这种程序也可以从数据处理器中读出。通过控制数据处理器操作，可以在数据处理器中实现该路由软件核心 4。

- 15 路由软件核心 4 的性能可以通过包括各种指令的程序来实现，并且通过计算机可读的记录介质来实现。

在说明书中，术语“记录介质”意指可以记录数据的任何介质。

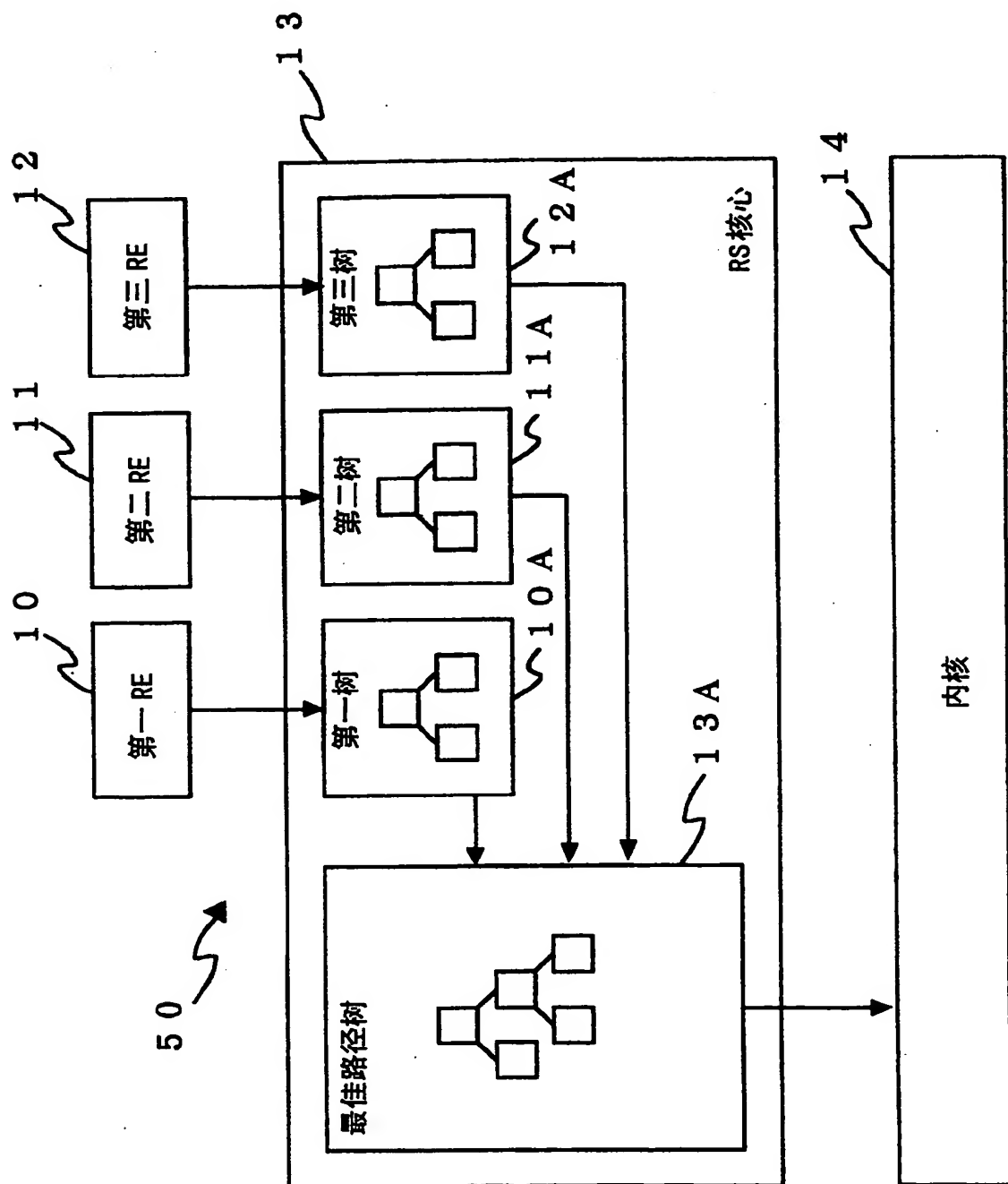


图1

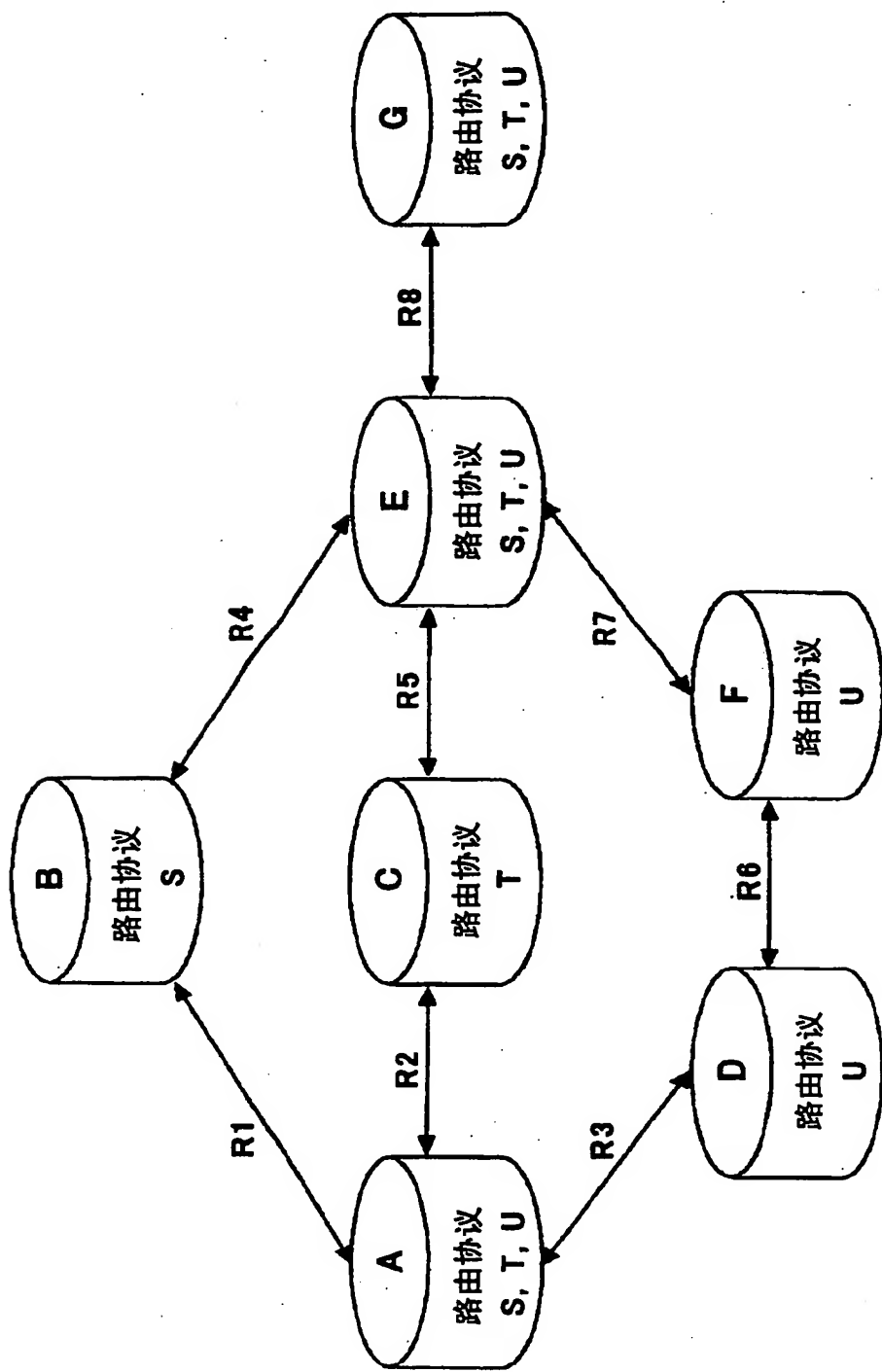


图 2

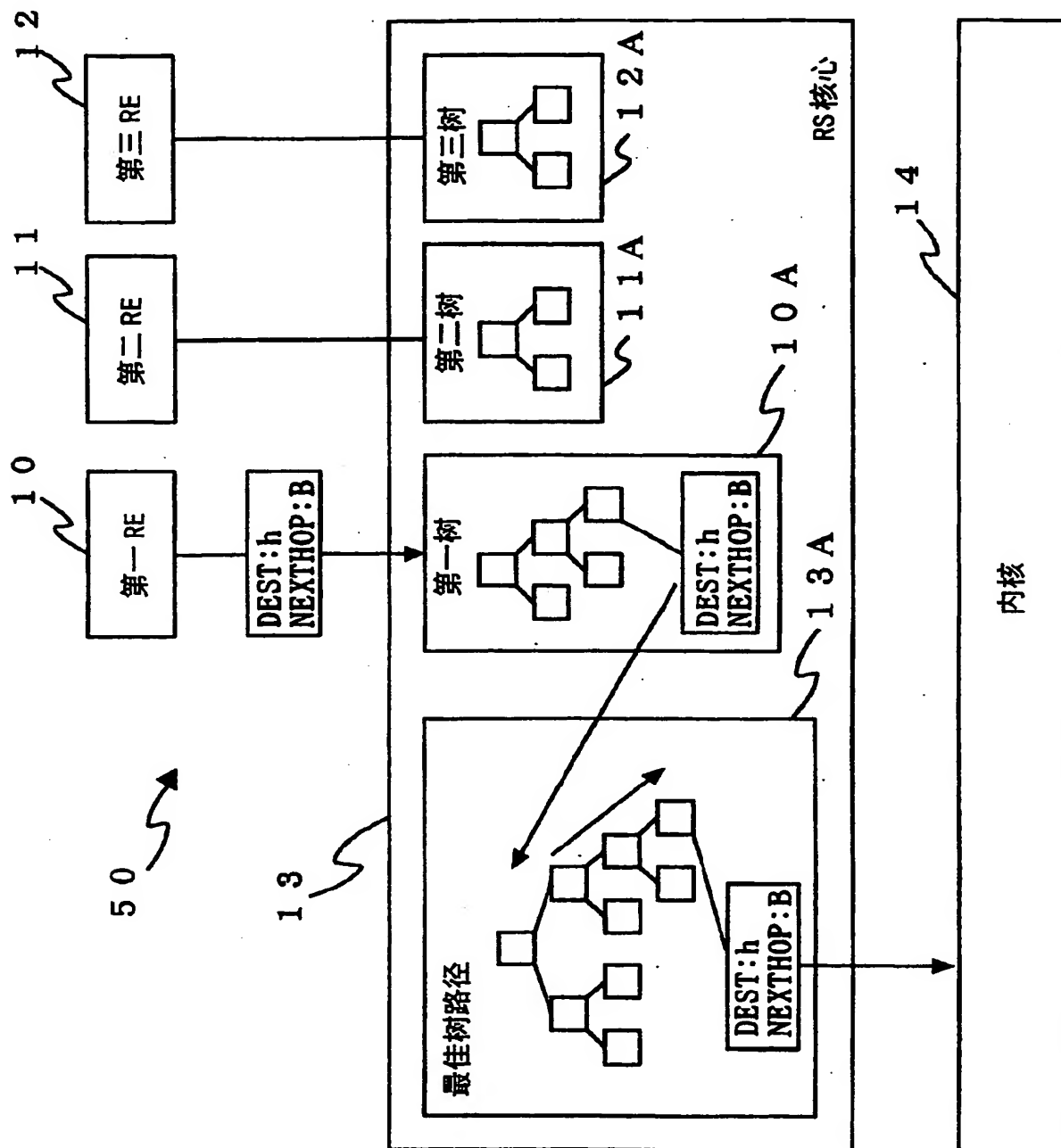


图 3

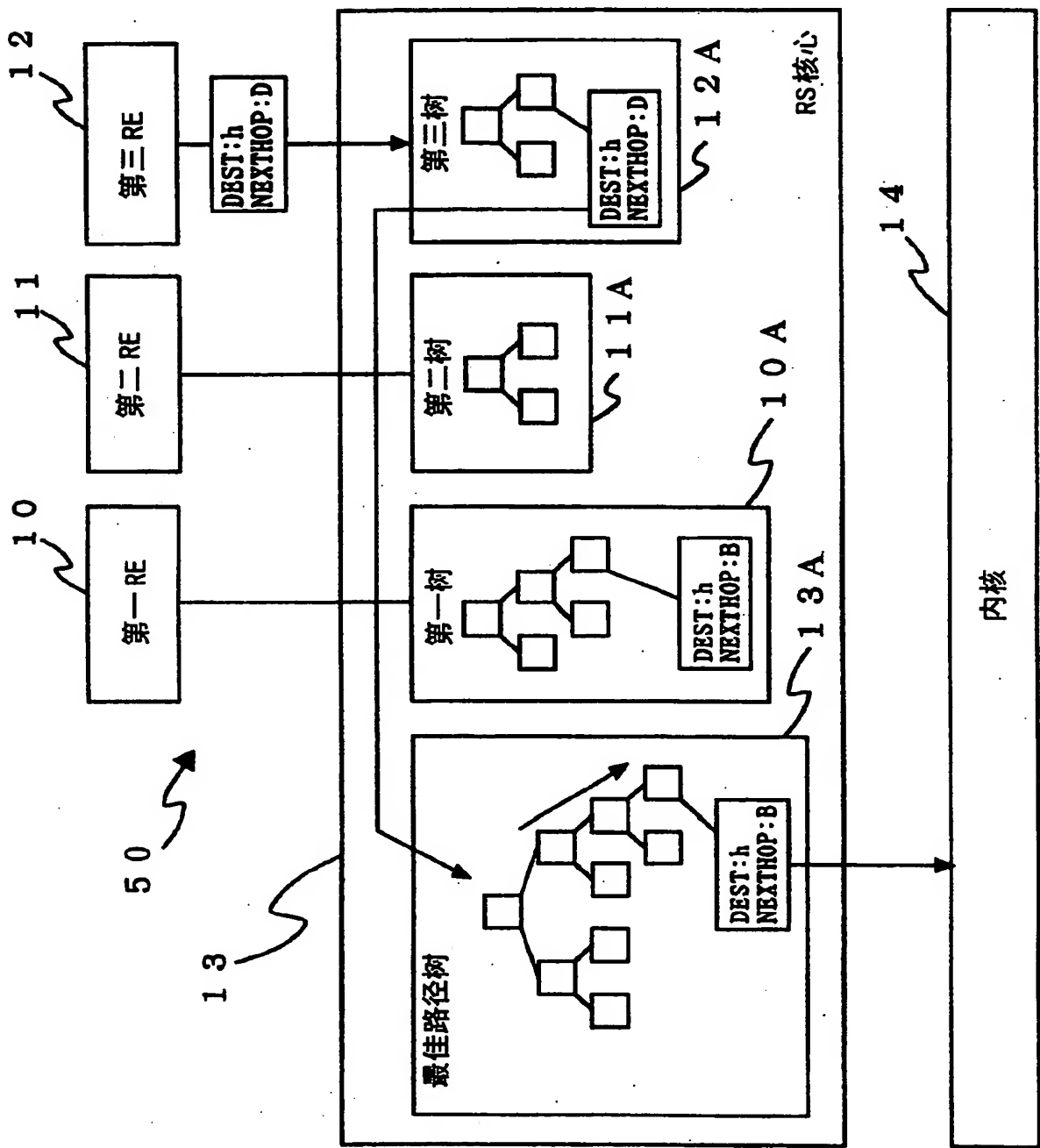


图 4

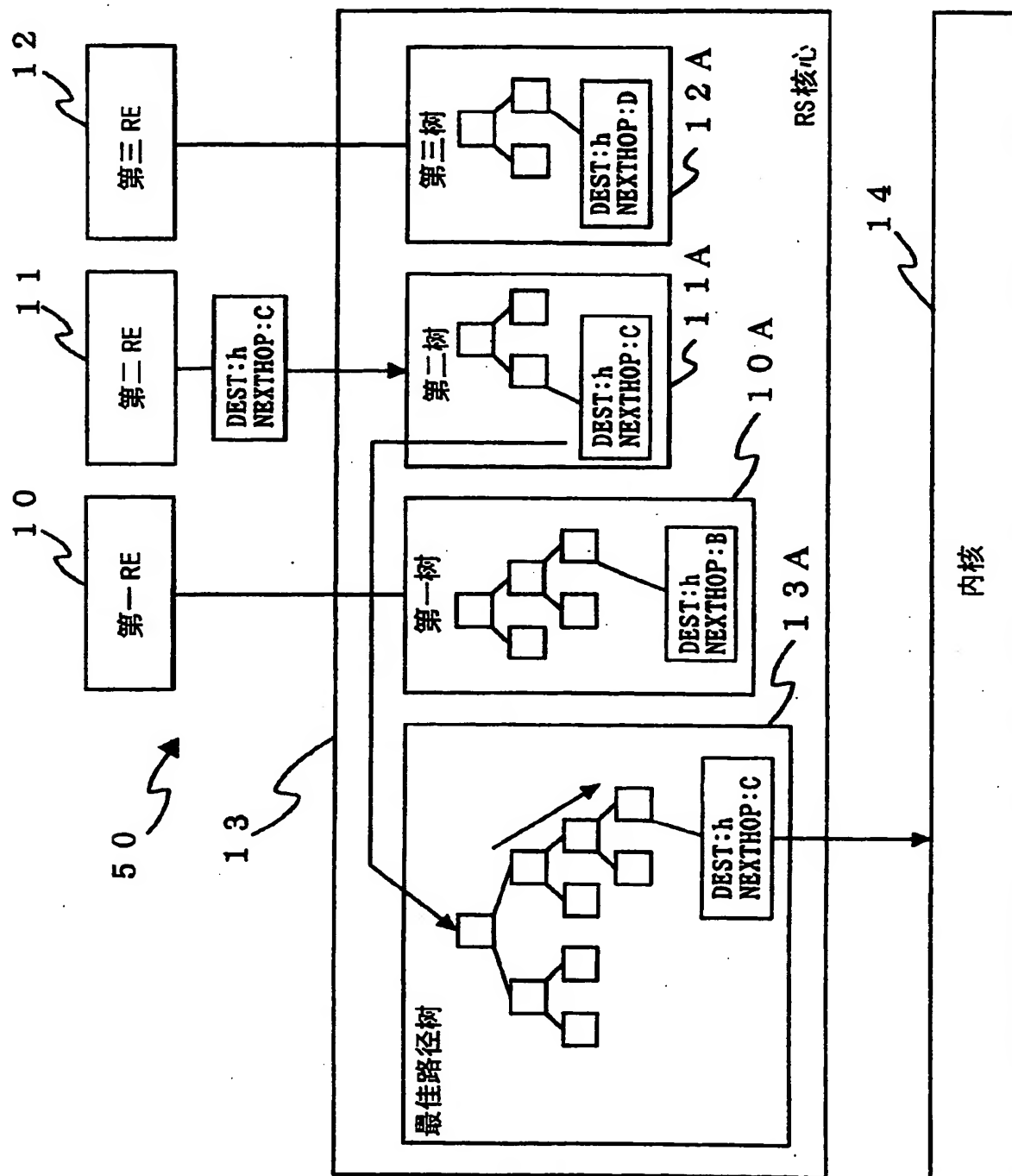


图 5

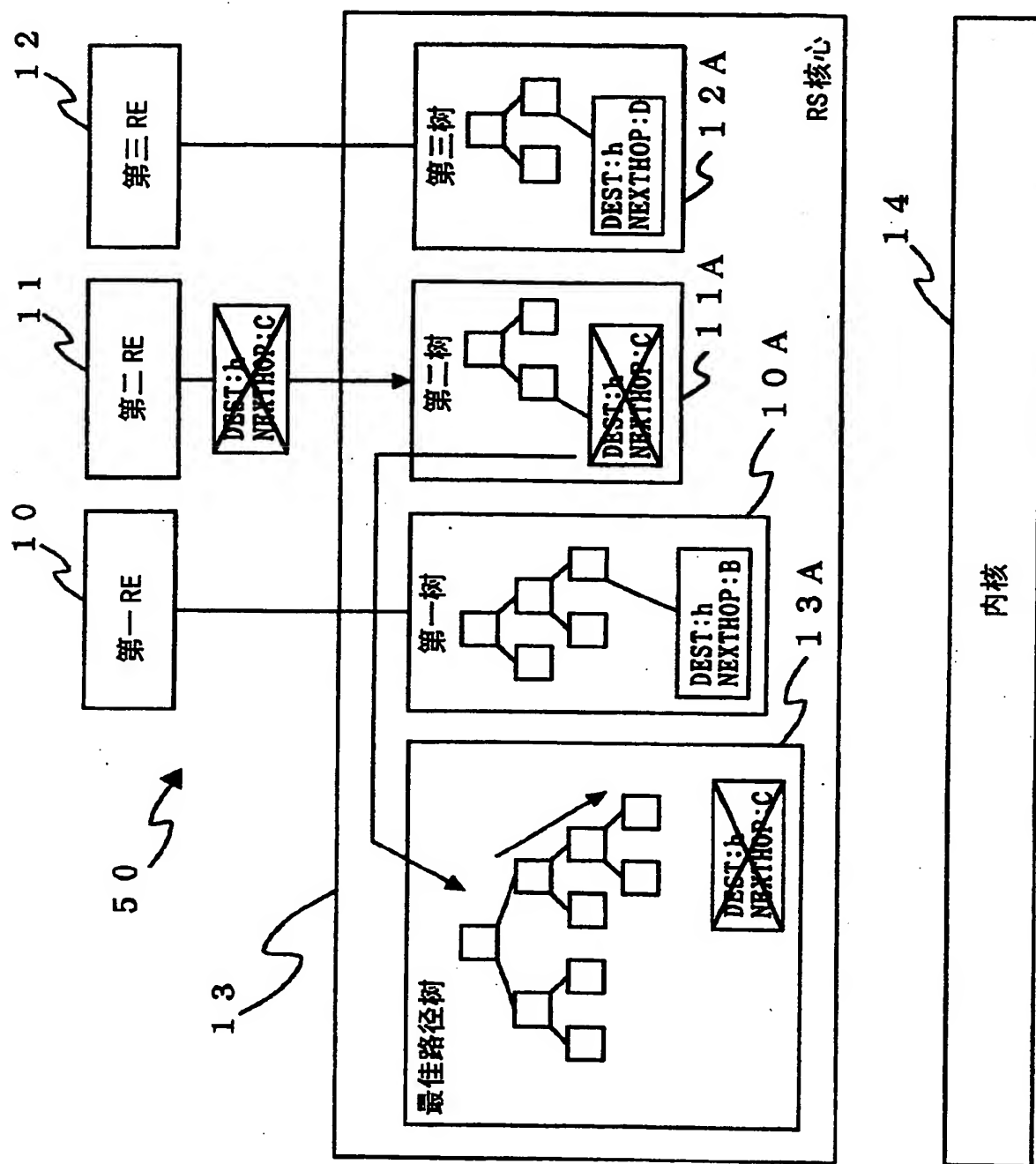


图 6

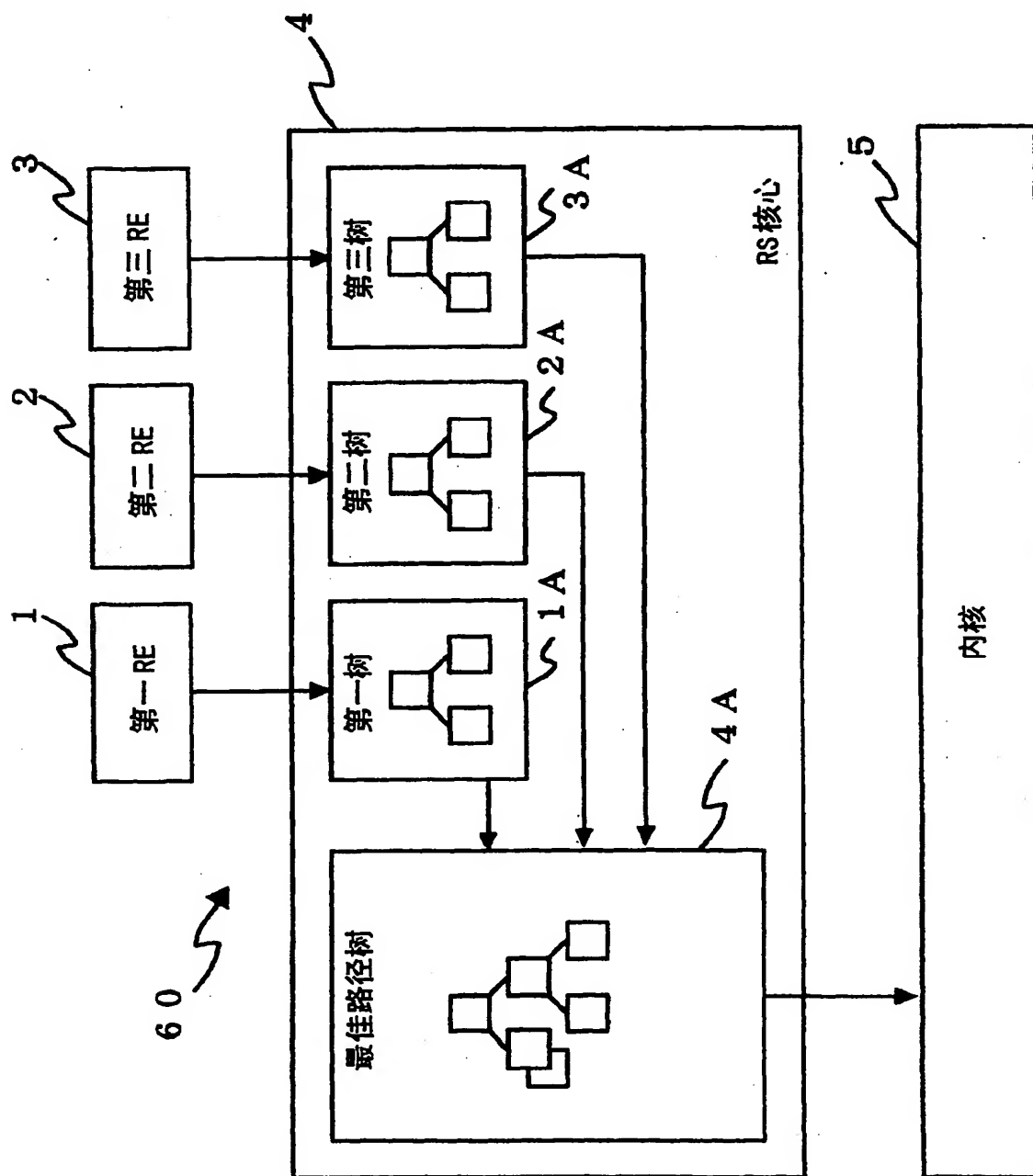


图 7

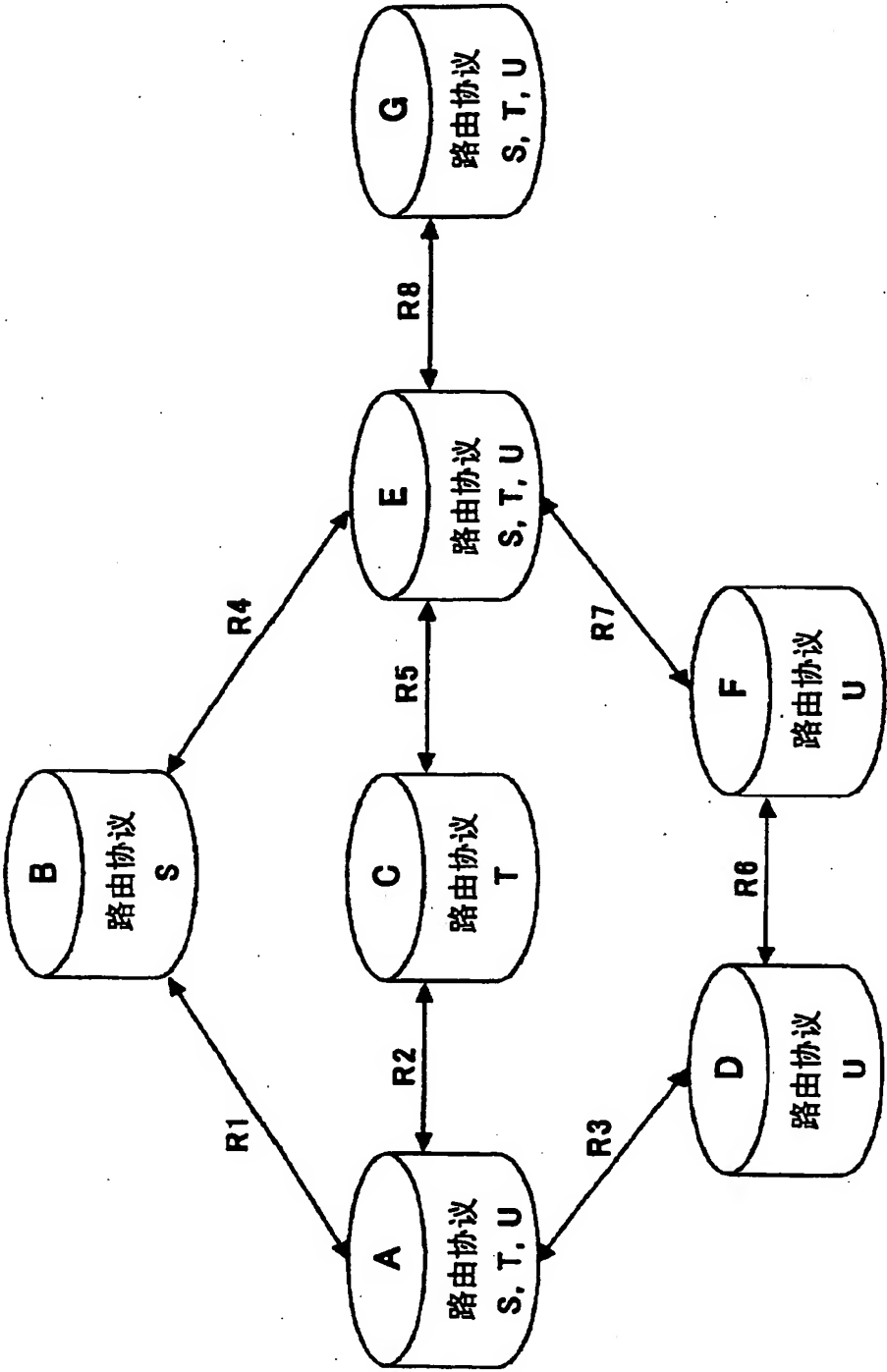


图 8

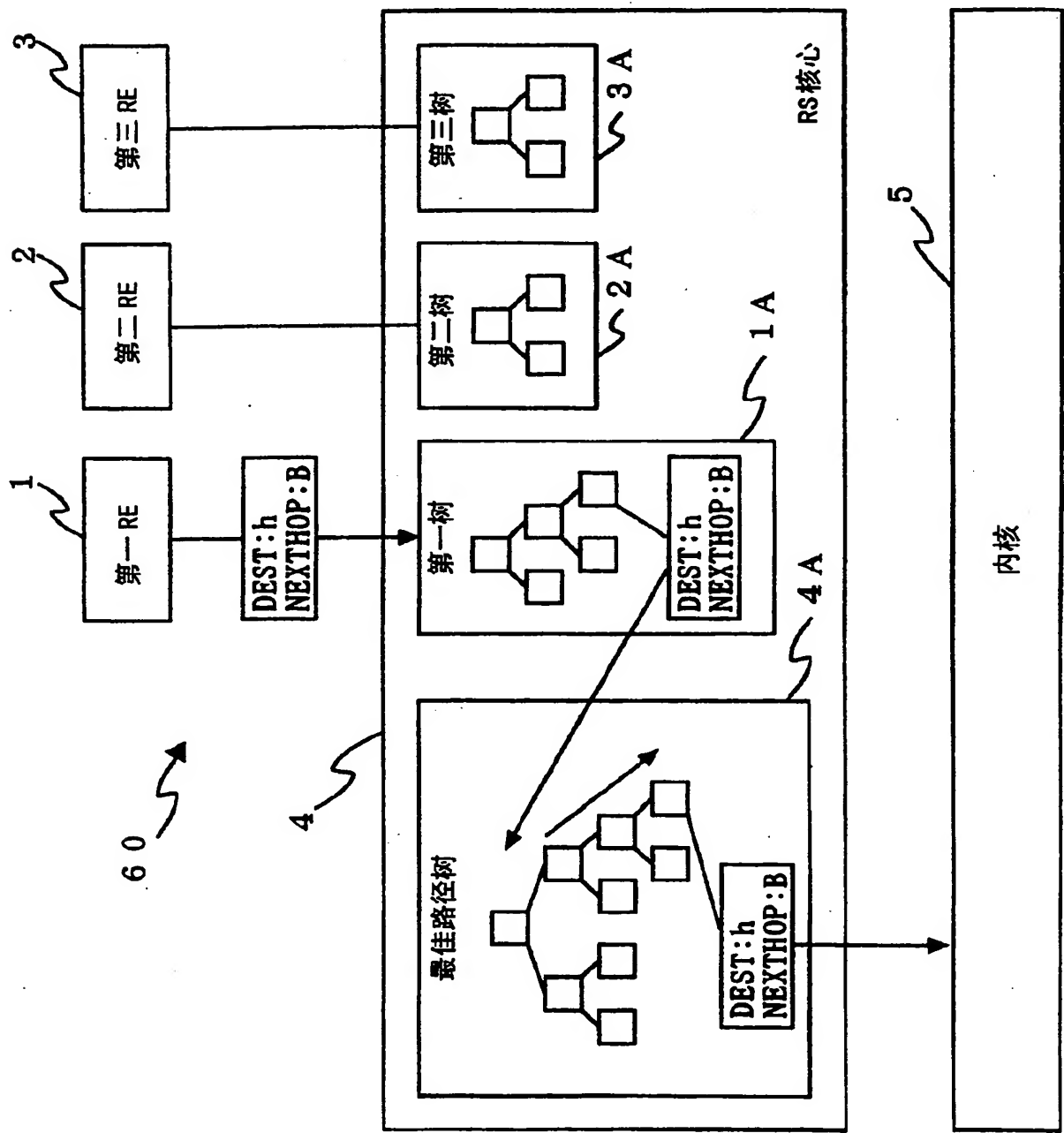


图 9

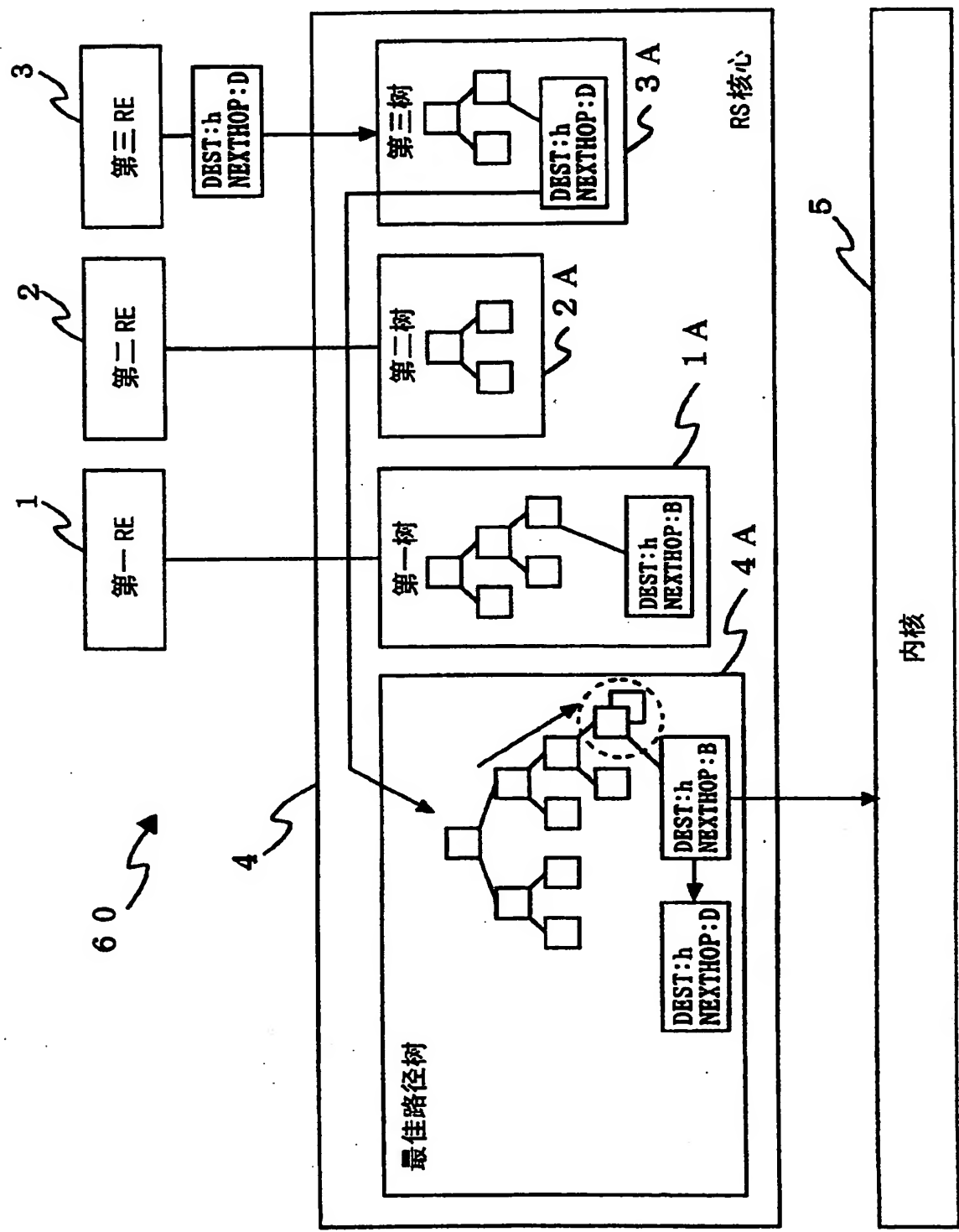


图 10

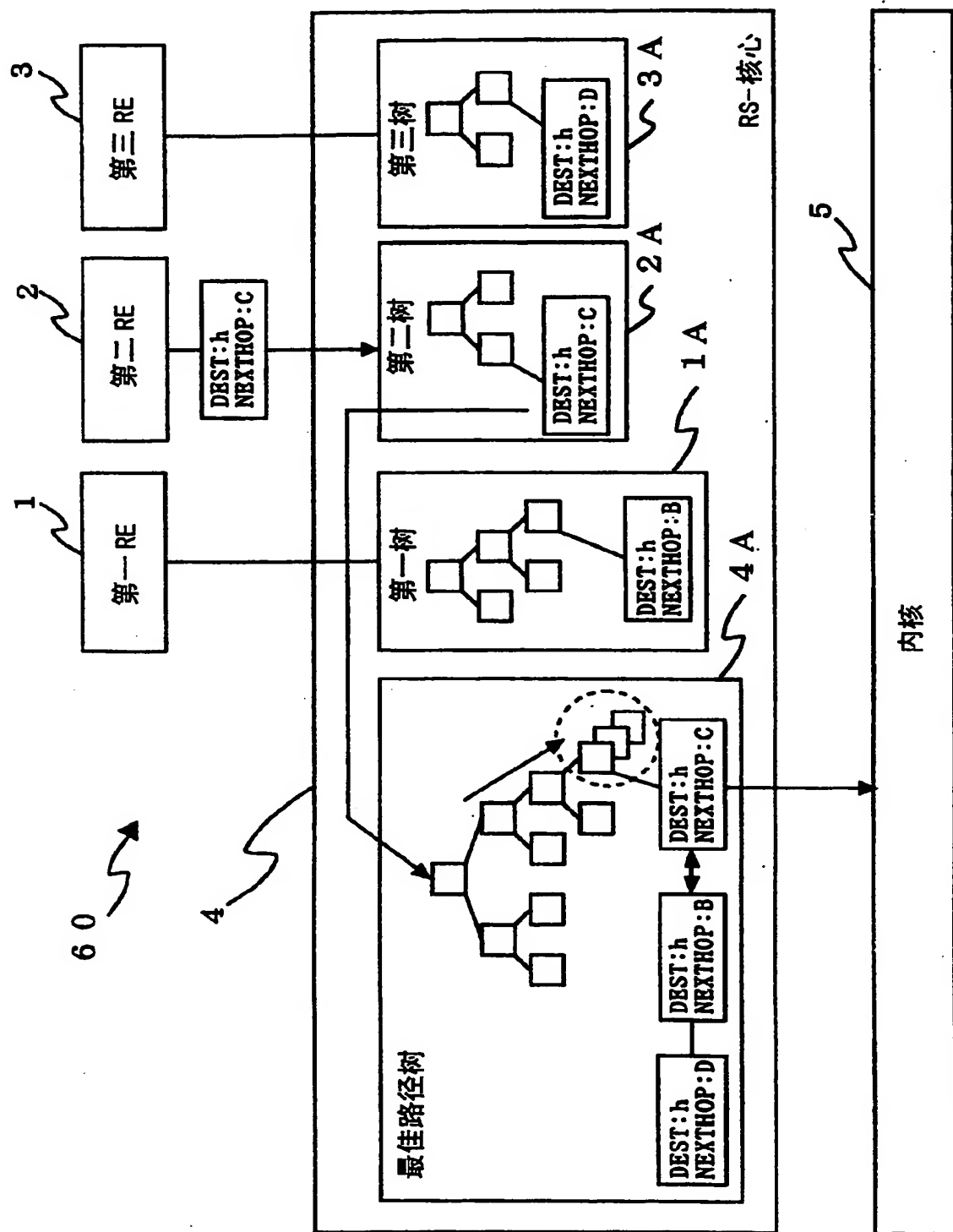


图 11

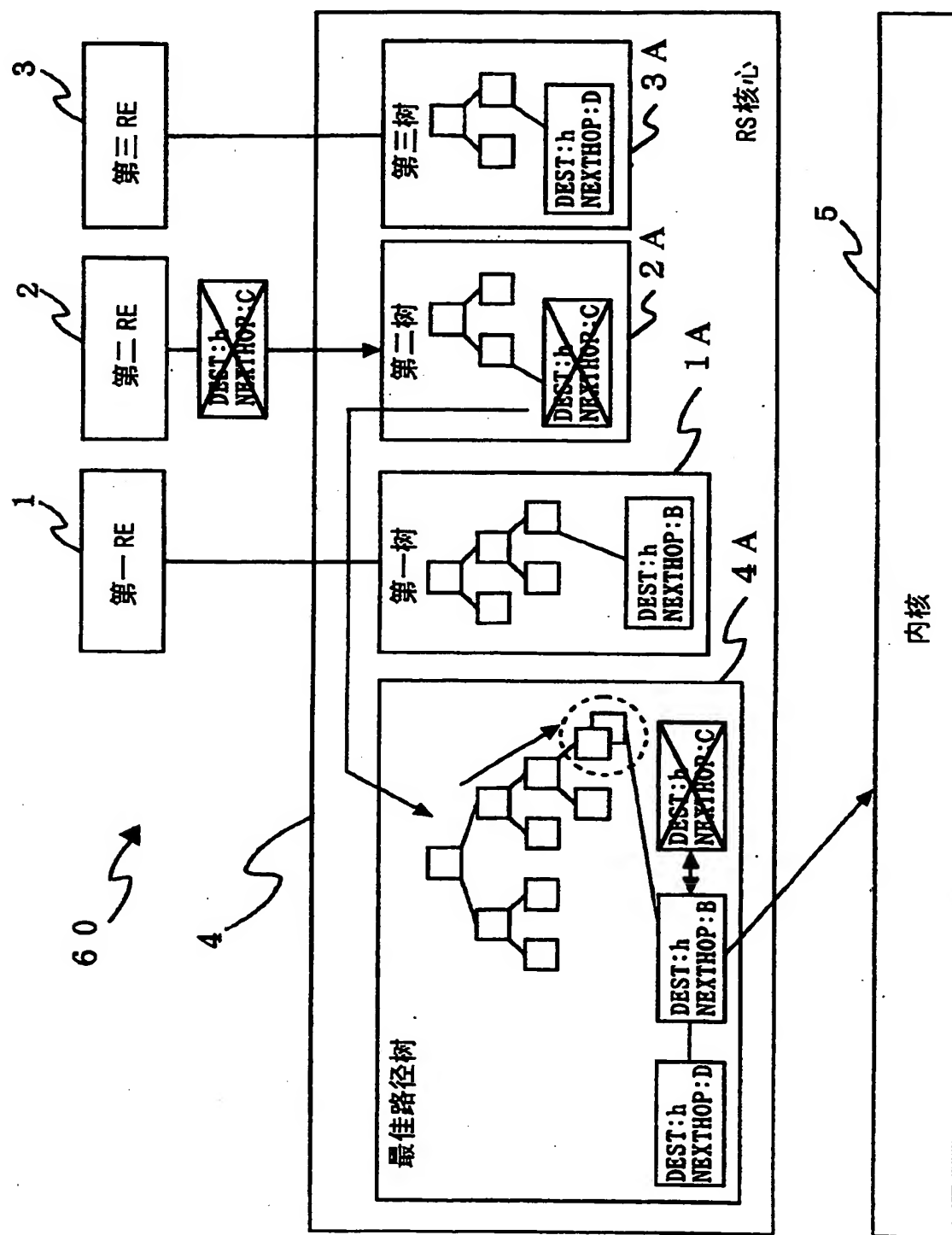


图 12